6 Benchmark Results

As mentioned in the system performance section of this paper, both single node and multi-node performance measurements are required in determining the overall performance of a distributed system. The following is a list showing single node and multi-node performance metrics and the associated benchmarks used to analyze the DAISy cluster.

Single Node Performance	*Associated Benchmarks*
raw CPU performance	*theoretical floating point performance*
1st level & 2nd level cache	*Matrix-Matrix multiply (using assembly coded dgemm)*
1st level & 2nd level cache & main memory	*LINPACK, SPEC, lmbench suite, STREAM*
Disk I/O	*Bonnie*
Network and Parallel Parallel Performance	
low level latency and bandwidth	*lmbench suite, Netperf suite*
API level latency and bandwidth	*PVM Timing Example, MPI Timing Example*
NFS performance	*Bonnie, cp*
application level parallel performance	*NPB (PVM & MPI), Parallel Seismic Inverse Problem*

For comparison, there are various other system performance results shown thoughout the paper. Table 1 shows a description of the DAISy and HEAT clusters. A note as to where the other system performance results were obtained will be given in the respective subsections. The measured MHz results are form the lmbench suite [3]. The HP machines appear to measure the clock speed at 1/3 speed.

System Description	Operating System	CPU	Measured MHz	Specified MHz	Year	List Price
FreeBSD/i586 (p5-90)	FreeBSD 2.1	Pentium 90	90.14	90	'94	5K
DEC Alpha	OSF1 3.2	Alpha	173.06	175	'92	24.4K
HP 9000/735	HP-UX A.09.05	PA-RISC	32.97	99	'92	33K
IBM RS6000	AIX 2.3.5	RS6000	40.85	40	'92	29K
SGI IRIX	IRIX 5.3	MIPs R4000	98.35	100	'92	33K
SUN SS10,SUN/TI	SUNOS 4.1.3	Super SPARC	49.75	50	'92	18.9K

Table 1. DAISy and HEAT Cluster System Descriptions. Measured MHz are from the lmbench suite [3].

6.1 Theoretical Floating-point Performance

The performance of the Pentium CPU on floating point is discussed in detail in [18],[19]. The following discussion, and performance measurements, are for 64 bit floating point operations. The Pentium CPU has a single 8 stage pipeline for floating point operations, with a capability of producing one result per cycle, so the "not-to-be-exceeded' performance is 1 FLOP per cycle, or 90 MFLOPS for the 90 MHz P54C. However, the P54C architecture uses a stack of floating point registers rather than an independently addressable register set. A common operation is to swap operands within the stack. When paired with certain floating point operations this swap can happen simultaneously, so that no cycles are lost. This cannot be done in every case. The most highly tuned assembly coded kernels achieve a maximum floating point performance of roughly 2/3 FLOP per cycle, or 60 MFLOPS for the 90 MHz P54C.

6.2 Matrix-Matrix Multiply (using assembly coded DGEMM)

Matrix-Matrix multiply is a highly optimized kernel computation which, when well implemented, is for practical purposes an indicator of the upper bound of the floating point performance available to user applications on a single node. The performance figures provided here use the BLAS3 [20] DGEMM algorithm. The BLAS 3 routines are generally provided as tuned assembly coded routines by RISC UNIX workstation vendors. The P54C-90 Pentium results were obtained using a tuned version of the DGEMM algorithm, using assembly coded DAXPY routines implemented by Russell Carter (contractor at Sandia, CA). The best performance obtained was 13.3 MFLOPS on 64x64x64 DGEMM. Properly tuned DGEMM implementations allow for the CPU to operate out of the highest level of the memory hierarchy, in this case, the first and second level cache.

6.3 LINPACK: Standard Linear Equations Software

The LINPACK [21] results reflect only one problem area: solving dense systems of equations. The LINPACK programs can be characterized as having a high percentage of floating -point arithmetic operations. Most floating-point operations in LINPACK take place in a set of subprograms, the Basic Linear Algebra Subprograms (BLAS), which are called repeatedly throughout the calculation. The results in Tables 2 and 3 show both C version and FORTRAN versions of LINPACK run on DAISy and HEAT. The FORTRAN versions are compiled with the FORTRAN to C option. Code is provided on the NETLIB [26] software repository for both C and FORTRAN versions.

linpackc	p5-90	DEC Alpha	HP 735	IBM RS6K	SGI IRIX	SUN SS10
MFLOPS	7.4	19.3	18.7	12.6	8.1	10.1

Table 2. LINPACK C results.

linpackd	p5-90	DEC Alpha	HP 735	IBM RS6K	SGI IRIX
MFLOPS	6.151	9.909	5.607	4.862	4.86

Table 3. LINPACK FORTRAN results.

6.4 Stream: A Main Memory Benchmark

McCalpin's Stream benchmark [22] is the most widely used measure of main memory bandwidth. The STREAM benchmark is specifically designed to work with datasets much larger than the available cache on any given system, so that the results are (presumably) more indicative of the performance of very large, vector style applications. Table 4 shows the main memory bandwidth results for the DAISy and HEAT systems with regard to the various function rates. This benchmark performs a carefully parameterized DAXPY, and returns information on several aspects of main memory performance. The benchmark is notable because it emphasizes the measurement of the rate at which operands can be fetched to the CPU from the lowest level of the memory hierarchy, which for most workstation class systems is the large DRAM based main memory. Appendix B gives an explanation of the various function rates.

	MB/s			
Function Rate	p5-90	DEC Alpha	IBM RS6K	SGI IRIX
Assignment:	38.908	88.148	125.912	39.903
Scaling :	39.178	88.83	122.966	37.597
Summing :	46.03	93.155	130.42	38.856
SAXPYing :	46.032	91.699	130.285	36.285

Table 4. Stream results (MB/s).

6.5 SPECx92: Standard Performance Evaluation Corporation

The **S**tandard **P**erformance **E**valuation **C**orporation (SPEC) [23] was founded in, 1988, as a non-profit group of computer vendors, system integrators, universities, research organizations, publishers and consultants throughout the world. It was formed with the objective to establish, maintain and endorse a standardized set of relevant benchmarks that can be applied to the newest generation of high-performance computers. The SPEC92 benchmarks are the second generation of the SPEC benchmarks. A third major version is the SPEC95 suite. Unfortunately, there were insufficient results reported for systems that resemble the DAISy systems for the SPEC95 suite. Below are various results from the SPECint92 and SPECfp92 results reported from the SPEC URL [1], *http://hpwww.epfl.ch/bench/SPEC.html* and [2] *http://www.ideas.com.au/bench/spec/spec.htm*.

SPECint92 is derived from the results of a set of integer benchmarks, and can be used to estimate a machine's single-tasking performance on integer code.

SPECfp92 is derived from the results of a set of floating point benchmarks, and can be used to estimate a machine's single-tasking performance on floating-point code.

Table 5 shows DAISy and HEATs individual system SPEC performance. Figure 15 is interesting in that the results show performance growth of microprocessors based upon Intel's 80x86 microprocessor architecture. Refer to the appendices for SPECint92 and SPECfp92 results showing the top 20 SPECx92 performance ratings as reported at [2], and the results of the P5-90 and P6-200. The P6-200 is the PentiumPro 200MHz CPU and may become part of the next generation DAISy cluster . Results show that the P6-200 CPU alone is 3.5 times faster than the current DAISy system CPUs in both integer and floating-point operations.

System Name	CPU (Num.)	ClkMHz Type et/in	Cache ext./D	SPECint 92	SPECfp 92	Inf. Date	Source Obtained
Intel Xpress	Pentium	60/90	512+8/8	106.5	81.4	Mar-95	www.intel
DEC 3000/600S	A21064	35/175	2M+8/8	114.1	162.1	Oct-93	c.arch
HP 7[35]5	PA7100	99	256/256	109.1	167.9	Jan-94	HP
SGI IndigoR4000	R4000	50/100	1M+8/8	57.6	60.3	Mar-93	c.sun.hw
Sun SS10/51	SuprSP	40/50	1M+20/16	65.2	83	Apr-93	Sunflash

Table 5. DAISy and HEAT individual systems SPEC performance

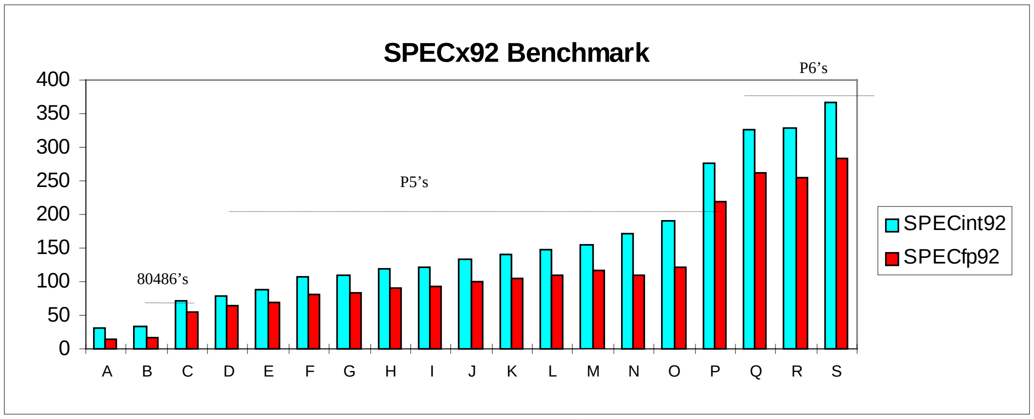

Figure 15. SPECint92 and SPECfp92 results showing
performance growth for CPUs designed after the 80x85 architecture [1].

6.6 lmbench: Portable Tools For Performance Analysis

The lmbench suite [2,3] is a set of micro-benchmarks intended for measuring system performance. It is intended for the computer community as a set of tools designed to focus attention on the basic building blocks of many common system applications, such as databases, simulation, software development, and networking. The lmbench suite attempts to measure the most commonly found performance bottlenecks. The suite is broken up into three categories: bandwidth, latency, and other. The identified bottlenecks are reproduced in the micro-benchmarks which measure system latency and bandwidth of data movement among the processor and memory, network, and file system. The other benchmark in the suite is a clock rate measurement and the measured results are shown in the DAISy and HEAT system descriptions at the beginning of this section.

The bandwidth measurements are intended to show how fast a system can move data. The ways that are measured are: the read() interface, library bcopy() and hand unrolled bcopy(), direct memory read and write (no copying), through the mmap() interface, through pipes, and TCP sockets. The bandwidth benchmarks are divided between three components: memory speed (memory bandwidth), operating system overhead (IPC bandwidth), and cache reusability (cached I/O bandwidth).

The latency measurements are intended to show how fast a system can be told to do an operation, or, the time between the start and the completion of an event. The latency measurement benchmarks included are: memory latency, basic operating system entry cost, process creation times, context switching, interprocess communication, file system latency, page fault latency, and memory mapping latency. For various interprocess communication tests, both local and remote hosts are tested.

Name Used	Vender & Model	Multi or Uni	Operating System	CPU	MHz	Year	List Price
IBM PowerPC	IBM 43P	Uni	AIX 3.?	MPC604	133	95	15K
IBM Power2	IBM 990	Uni	AIX 4.?	Power2	71	93	110K
FreeBSD/i586	ASUS P55TP4XE	Uni	FreeBSD 2.1	Pentium	133	95	6K
HP K210	HP 9000/859	MP	HP-UX B.10.01	PA 7200	120	95	35K
SGI Challenge	SGI Challenge	MP	IRIX 6.2-alpha	R4400	200	94	80K
SGI Indigo2	SGI Indigo	Uni	IRIX 5.3	R4400	200	94	15K
Linux/Alpha	DEC Cabriolet	Uni	Linux 1.3.38	Alpha 21064A	275	94	9K
Linux/i586	Triton/EDO RAM	Uni	Linux 1.3.28	Pentium	120	95	5K
Linux/i686	Intel Alder	Uni	Linux 1.3.37	PentiumPro	200	95	7K
DEC Alpha@150	DEC 3000/500	Uni	OSF1 3.0	Alpha 21064	150	93	35K
DEC Alpha@300	DEC 8400 5/300	MP	OSF1 3.2	Alpha 21164	300	95	?250K
Sun Ultra1	Sun Ultra	Uni	SunOS 5.5	UltraSparc	167	95	21K
Sun SC1000	Sun SC1000	MP	SunOS 5.5-beta	SuperSparc	50	92	35K
Solaris/i686	Intel Alder	Uni	SunOS 5.5.1	Pentium Pro	133	95	5K
Unixware/i686	Intel	Uni	Unixware 5.4.2	Pentium Pro	200	95	7K

Table 6. System descriptions from the *lmbench Document* [3].

The results from the HEAT and DAISy clusters will be condensed and compared with results from *Larry McVoy and Carl Staelin's January, 1996 lmbench Document* [3]. Table 6 shows the System descriptions from the *lmbench Draft.* The results from the DAISy and HEAT clusters are in full in the appendices.

6.6.1 Bandwidth Measurements

6.6.1.1 Memory Bandwidth

The lmbench suite measures the ability to copy, read, and write data over a varying set of sizes. The benchmarks included in the *memory bandwidth* component are: bw_mem_cp, bw_mem_rd, and bw_mem_wr. The results are shown in Table 7.

System	bcopy (bw_mem_cp) unrolled	libc	memory read (bw_mem_rd)	write (bw_mem_wr)
IBM Power2	242	171	205	364
Sun Ultra 1	85	167	129	152
DEC Alpha@300	85	80	120	123
HP K210	78	57	117	126
Unixware/i686	65	55	214	86
DEC Alpha @150	46	45	79	91
Linux/i686	42	57	208	56
Linux/Alpha	39	39	73	71
FreeBSD/i586	39	42	73	83
Linux/Alpha	39	39	73	71
Linux/i586	38	42	74	75
SGI Challenge	35	36	65	67
SGI Indigo	31	32	69	66
IBM PowerPC	21	21	63	26
Sun SC1000	17	15	38	31
DAISY systems				
FreeBSD/i586 (p5-90)	19.59	19.59	54.15	28.01
HEAT systems				
DEC Alpha	45.63	45.11	84.8	90.31
HP 9000/735	25.86	25.97	52.81	50.88
IBM RS6000	61.15	45.65	59.03	61.32
SUN SS10	18.27	16.99	42.6	28.18
SGI IRIX	22.93	23.27	44.47	48.94

Table 7. bw_mem_cp, bw_mem_rd, bw_mem_wr results (MB/s).

Copy bandwidth is measured two ways. The first is the user-level library *bcopy* interface. The second is a hand-unrolled loop that loads and stores 8-byte words. The results from our tests denote both aligned and unaligned stores of the 8 byte words. For our HEAT and DAISy cluster tests unaligned stores showed better results and are the only ones documented, from the *McVoy* and *Staelin* draft aligned stores are documented. In order to show memory bandwidth rather than cache bandwidth, only the results that copy an 8MB area to another 8MB area are shown. In the appendices the bw_mem_cp results show the cache bandwidth as well as memory bandwidth results. As the copy size increases the cache bandwidth is negated. For the *McVoy* and *Staelin* results: Some of the PCs had less than 16MB of available memory; those machines copied 4MB.

Memory reading bandwidth is measured by an unrolled loop that sums up a series of integers (typically a 4 byte integer). The benchmark bw_mem_rd allocates the specified amount of memory, zeros it, and then times the reading of the memory as a series of integer loads and adds. Basically, a large array is repeatedly read sequentially. Again, an 8MB area is specified to show memory bandwidth and not cache bandwidth. In the appendices the bw_mem_rd results show the cache bandwidth as well as memory bandwidth.

Memory writing bandwidth is measured by an unrolled loop that stores a value into an integer (typically a 4 byte integer) and then increments the pointer. The benchmark bw_mem_wr allocates the specified amount of memory, zeros it, and then times the writing of that memory as a series of integer stores and increments. Basically, a large array is repeatedly written sequentially. Again, an 8MB area is specified to show memory bandwidth and not cache bandwidth. In the appendices the bw_mem_wr results show the cache bandwidth as well as memory bandwidth.